

Unisenac
Centro Universitário RS



Fecomércio
Sesc

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Algoritmos e Estruturas de Dados I

PROF. EDÉCIO FERNANDO IEPSSEN

Manipulação de Strings

- ▶ Diversas são as operações, em programação, que necessitam manipular cadeias de caracteres (strings).
- ▶ São úteis, por exemplo, para validar uma senha, gerar uma sugestão de e-mail, extrair palavras de um texto, dividir uma linha em partes.
- ▶ Todas as linguagens de programação dispõe de métodos específicos para trabalhar com strings, o que muda de uma para a outra, é a sintaxe.
- ▶ Uma cadeia de caracteres é uma sequência de letras, algarismos ou símbolos (símbolos de pontuação, parênteses, etc.).

Funções Python para Manipulação de Strings

```
cidade="Pelotas"      # exemplo de variável string com o conteúdo "Pelotas"
len(cidade)           # retorna o tamanho da string: 7
cidade.upper()        # converte para maiúsculas: "PELOTAS"
cidade.lower()        # converte para minúsculas: "pelotas"
cidade.find("t")      # encontra a posição de "t" em "Pelotas": 4. Obs.: se não existir
retorna -1
cidade.rfind("e")     # pesquisa do final para o início da string
cidade.count("t")     # conta o número de ocorrências do caracter (ou caracteres) na string: 1
split()              # cria um vetor a partir de uma string, utilizando um caracter de
separação.
cidade="Santa Vitória do Palmar"
partes=cidade.split(" ") # partes[0] = "Santa"; partes[1] = "Vitória"; partes[2] = "do"; ...
```

Obs.: As strings em Python são “imutáveis”, ou seja, não é possível alterar parte do seu conteúdo. Desta forma, operações como replace ou lower não podem ser realizadas sobre a

Operações sobre Strings

```
a = "Redes"
b = a * 4
print(b)                                # RedesRedesRedesRedes
print("-" * 10)                          # -----
print(a.center(20,"*"))                  # *****Redes*****
print(a.ljust(20,"*"))                   # Redes*****
print(a.rjust(20,"*"))                   # *****Redes
```

Alinhamentos de Strings no print(f" ")

O significado das várias opções de alinhamento é o seguinte:

Opção	Significado
'<'	Força o alinhamento à esquerda do campo dentro do espaço disponível (este é o padrão para a maioria dos objetos).
'>'	Força o alinhamento à direita do campo dentro do espaço disponível (este é o padrão para números).
'='	Força o preenchimento a ser colocado após o sinal (se houver), mas antes dos dígitos. É usado para imprimir campos na forma "+000000120". Esta opção de alinhamento só é válida para tipos numéricos. Torna-se o padrão para números quando "0" precede imediatamente a largura do campo.
'^'	Força a centralização do campo no espaço disponível.

Alinhamentos de Strings no print(f" ")

```
C:\Users\edeci>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35)
Type "help", "copyright", "credits" or "license" for more ir
>>> a = "oi"
>>> print(f"{a:10}x")
oi          x
>>> print(f"{a:>10}x")
          oix
>>> print(f"{a:^10}x")
      oi      x
>>> print(f"{a:*^10}x")
****oi****x
```

Indexação e fracionamento

Em Python (e na maioria das linguagens de programação), uma string pode ser manipulada como um vetor de caracteres. Assim, caso a variável **cidade = "Pelotas"** se quisermos obter o primeiro caracter desta string podemos referenciar:

```
print(cidade[0]) # irá apresentar "P"
```

Fracionamento:

```
print(cidade[0:3]) # irá apresentar "Pel".
```

```
print(cidade[2:4]) # irá apresentar "lo"
```

```
print(cidade[2:]) # da posição 2 até o final. Irá apresentar "lotas"
```

```
print(cidade[:2]) # os 2 primeiros caracteres. Irá apresentar "Pe"
```

```
print(cidade[-1]) # do final para o início. Irá apresentar o último  
caracter
```

Verificação dos caracteres de uma String

Exemplos:

```
letra = "a"
```

```
letra.isalpha()      # verifica se é alfabético
```

```
letra.isdigit()      # verifica se é dígito numérico
```

```
letra.islower()      # verifica se é minúsculo
```

```
letra.isupper()      # verifica se é maiúsculo
```

```
letra.isspace()      # verifica se é espaço
```

<https://realpython.com/python-strings/>